Fast Learning of On-line EM Algorithm

Masa-aki Sato
ATR Human Information Processing Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
TEL: (+81) 774-95-1039 FAX: (+81) 774-95-1008
E-mail: masaaki@hip.atr.co.jp

Abstract

In this article, an on-line EM algorithm is derived for general Exponential Family models with Hidden variables (EFH models). It is proven that the on-line EM algorithm is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix as a coefficient matrix. As a result, the stochastic approximation theory guarantees the convergence to a local maximum of the likelihood function.

The performance of the on-line EM algorithm is examined by using the mixture of Gaussian model, which is a special type of the EFH model. The simulation results show that the on-line EM algorithm is much faster than the batch EM algorithm and the on-line gradient ascent algorithm. The fast learning speed is achieved by the systematic design of the learning rate schedule. Moreover, it is shown that the on-line EM algorithm can escape from a local maximum of the likelihood function in the early training phase, even when the batch EM algorithm is trapped to a local maximum solution.

Keywords

On-line algorithm, EM algorithm, Convergence, Mixture models, Exponential family models

1 Introduction

In recent years, the EM algorithm (Dempster, Laird and Rubin 1977) has been applied to a number of neural network models (Jordan & Jacobs 1994; Amari 1995; Xu et al. 1995). The EM algorithm is a general method for finding a maximum likelihood estimator for a stochastic model with hidden variables. Many neural network models have corresponding stochastic models (Amari 1995; Bishop 1995). The stochastic formulation helps to study the learning problem of neural networks. Hidden variables in the stochastic models correspond to hidden units in the neural networks, and these variables expand the representational capability of the models.

The EM algorithm is a batch algorithm in which all the training data are used to update model parameters at each iteration. It was proven that the likelihood for the training data always increases (or does not change) at each iteration of the EM algorithm (Dempster et al. 1977). Therefore, the EM algorithm guarantees the convergence to a local maximum of the likelihood function.

In real world problems, it is often desirable to use an on-line learning algorithm in which the training data are supplied one by one and the model parameters are updated each time using the current data. The most popular on-line algorithm is the on-line gradient descent algorithm. This algorithm can be considered as a stochastic approximation method (Robbins & Monro 1951) for finding a local minimum of an error function (Amari 1967; Kushner & Yin 1997; Bottou 1999; Murata 1999). On-line EM algorithms have also been proposed by several authors (Jordan & Jacobs 1994 ; Nowlan 1991). However, there has been no theoretical study on the convergence of the on-line EM algorithms. Neal and Hinton (1998) proposed a wide variety of incremental EM algorithms and proved their convergence. However, one drawback of their study is that their algorithms either need additional storage variables for all of the training data or need to see all of the training data at each iteration. This prevents these algorithms from being applied to real-time on-line settings where new data are indefinitely supplied each time.

In our previous paper, we proved the convergence of the on-line EM algorithm for a normalized Gaussian network (Sato & Ishii 1998). In this paper, we study more general models, namely the Exponential Family models with Hidden variables (EFH models), and derive an on-line EM algorithm for them. We prove that the on-line EM algorithm is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix as a coefficient matrix. As a result, the stochastic approximation theory guarantees the convergence to a local maximum of the likelihood function.

The performance of the on-line EM algorithm is examined by using the mixture of Gaussian model, which is a special type of the EFH model. The simulation results show that the on-line EM algorithm is much faster than the batch EM algorithm and the on-line gradient ascent algorithm. The fast learning speed is achieved by the systematic design of the learning rate schedule. Moreover, it is shown that the on-line EM algorithm can escape from a local maximum of the likelihood function in the early training phase, even when the batch EM algorithm is trapped to a local maximum solution.

The paper is organized as follows. Section 2 explains three kinds of learning problems for stochastic models. In this paper, only unsupervised learning problem is studied because the extension of the current method to the two kinds of supervised learning problems is obvious. Section 3 is a short review of the EFH model. The on-line EM algorithm for the EFH model is derived in Section 4. The equivalence of the on-line EM algorithm to the stochastic gradient method is proven in Section 5. Section 6 explains the systematic design of the discount factor schedule. Section 7 explains experimental results and Section 8 is a conclusion of this paper.

2 Learning Problem

In this section, three kinds of learning problems are defined for stochastic models.

2.1 Unsupervised Learning

For a set of observed input data $X \equiv \{x(t)|t = 1, ..., T\}$, unsupervised learning can be defined as a problem to find the stochastic model that best explains the observed input data. A stochastic model for the input distribution is defined by a probability distribution for an input x, $P(x|\theta)$, where θ represents a set of model parameters. Unsupervised learning for the stochastic model can be solved by the maximum likelihood method, which maximizes the log-likelihood for a set of observed data X, $\sum_{t=1}^{T} \log P(x(t)|\theta)$. This model tries to learn the input data distribution $\rho(x)$. The mixture of Gaussian model is an example of this type.

2.2 Supervised Learning

Deterministic neural networks are general purpose function approximators with adjustable (weight) parameters (Bishop 1995). For a set of observed input data $X \equiv \{x(t)|t = 1, ..., T\}$ and the corresponding target data (teacher signal) $Y \equiv \{y(t)|t = 1, ..., T\}$, supervised learning can be defined as a problem to find the optimal model parameter that minimizes some error function for a set of observed data (X, Y).

Many neural network models have corresponding stochastic models (Amari 1995; Bishop 1995). There are two kinds of supervised learning for stochastic models. The first kind of supervised learning defines a stochastic model by a conditional probability for an output y given an input x , $P(y|x, \theta)$. Supervised learning can be defined as a problem to find the optimal parameter that maximizes the log-likelihood of the conditional probability for a set of observed data (X, Y), $\sum_{t=1}^{T} \log P(y(t)|x(t), \theta)$. This model tries to learn the input-output relationship. In this case, the model does not take into account the input data distribution. Mixtures of expert models are examples of this type (Jacobs et al. 1991; Jordan & Jacobs 1994).

After learning, a deterministic output for a given input x can be calculated as the expectation value of the output:

$$\hat{y} = \int d\mu(y) y P(y|x, \theta), \tag{1}$$

where $d\mu(y)$ denotes a measure on the output space.

The second kind of supervised learning defines a model by a joint probability for an input x and an output y, $P(x, y|\theta)$. Here, supervised learning can be defined as a problem to find the optimal parameter that maximizes the log-likelihood of the joint probability, $\sum_{t=1}^{T} \log P(x(t), y(t)|\theta)$. This model tries to learn the joint input-output distribution. Therefore, the model takes into account the input data distribution as well as the input-output relationship. The normalized Gaussian network (Sato & Ishii 1998) is an example of this type. The conditional probability of the model can be calculated as

$$P(y|x, \theta) = P(x, y|, \theta) / \int d\mu(y') P(x, y'|\theta). \tag{2}$$

The deterministic output can be calculated by using Eq. (1) and (2).

One can define the Exponential Family models with Hidden variables (EFH models) for the above three types of stochastic models and derive an on-line EM algorithm for each type. In this paper, however, we only study the on-line EM algorithm for unsupervised learning. This is done purely for simplicity. It is obvious that the on-line EM algorithm for the two kinds of supervised learning can be derived in the same way as in this paper and that one can also prove the convergence of the on-line EM algorithm for these models.

3 Exponential Family model with Hidden variables

Let us define an Exponential Family model with Hidden variables (EFH model) for the input data distribution (Amari 1985). The input variable $\mathbf{x} \equiv (x_1, ..., x_D)^T$ could be a vector in the D-dimensional Euclidean space or could be a discrete variable vector. The hidden variable is assumed to be a discrete variable vector and is denoted by $\mathbf{z} \equiv (z_1, ..., z_L)^T$. A complete event of the model is specified by $(\mathbf{x}, \mathbf{z})$. An observed event $\mathbf{x}$ is called an incomplete event (Dempster et al. 1977). An EFH model is defined by a probability distribution for an observed event $\mathbf{x}$:

$$P(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\{\mathbf{z}\}} P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}), \tag{3a}$$

$$P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \exp[\sum_{i=1}^{M} r_i(\mathbf{x}, \mathbf{z})\theta_i + r_{M+1}(\mathbf{x}, \mathbf{z}) - \Psi(\boldsymbol{\theta})], \tag{3b}$$

where $\boldsymbol{\theta} \equiv (\theta_1, ..., \theta_M)^T$ denotes a set of model parameters and $\sum_{\{\mathbf{z}\}}$ denotes a sum over possible configurations for $\mathbf{z}$. $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ in Eq. (3) represents the probability distribution for a complete event $(\mathbf{x}, \mathbf{z})$. A set of sufficient statistics is defined by

$$\mathbf{r}(\mathbf{x}, \mathbf{z}) \equiv (r_1(\mathbf{x}, \mathbf{z}), ..., r_M(\mathbf{x}, \mathbf{z}))^T. \tag{4}$$

The normalization factor $\Psi(\boldsymbol{\theta})$ is determined by

$$\exp[\Psi(\boldsymbol{\theta})] = \int d\mu(\mathbf{x}) \sum_{\{\mathbf{z}\}} \exp[\mathbf{r}^T(\mathbf{x}, \mathbf{z}) \cdot \boldsymbol{\theta} + r_{M+1}(\mathbf{x}, \mathbf{z})], \tag{5}$$

where $d\mu(\mathbf{x})$ is a measure on the input space and $\mathbf{r}^T \cdot \boldsymbol{\theta} \equiv \sum_{i=1}^{M} r_i\theta_i$. Equation (5) is derived from a probability condition:

$$\int d\mu(\mathbf{x}) \sum_{\{\mathbf{z}\}} P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = 1. \tag{6}$$

The expectation parameter for the EFH model, $\boldsymbol{\phi} \equiv (\phi_1, ..., \phi_M)^T$, (Amari 1985) is defined by

$$\boldsymbol{\phi} \equiv \partial\Psi(\boldsymbol{\theta})/\partial\boldsymbol{\theta} \equiv (\partial\Psi/\partial\theta_1, ..., \partial\Psi/\partial\theta_M)^T \tag{7a}$$

$$= E[\mathbf{r}(\mathbf{x}, \mathbf{z})|\boldsymbol{\theta}] \equiv \int d\mu(\mathbf{x}) \sum_{\{\mathbf{z}\}} \mathbf{r}(\mathbf{x}, \mathbf{z}) P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}). \tag{7b}$$

It is known that the transformation between $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ is one-to-one. Therefore, Eq. (7) can be inverted as

$$\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\phi}). \tag{8}$$

The expectation parameter $\boldsymbol{\phi}$ can be used to parameterize the probability distributions of the EFH model and it is a dual set of the model parameter $\boldsymbol{\theta}$ (Amari 1985). By using the negative of the entropy,

$$H(\boldsymbol{\phi}) \equiv \int d\mu(\mathbf{x}) \sum_{\{\mathbf{z}\}} P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi})) \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi})) \tag{9a}$$

$$= \boldsymbol{\phi}^T \cdot \boldsymbol{\theta} - \Psi(\boldsymbol{\theta}), \tag{9b}$$

the model parameter $\boldsymbol{\theta}$ can be explicitly expressed as

$$\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\phi}) = \partial H(\boldsymbol{\phi})/\partial\boldsymbol{\phi}. \tag{10}$$

Equations (7), (9) and (10) form the Legendre transformation.

The Mixture of Exponential Family (MEF) model is a special type of the EFH models. The MEF model consists of N units. It is assumed that a single unit is selected from a set of N units for each observed event $\mathbf{x}$. The hidden variable of the MEF model is an indicator variable, $\mathbf{z} \equiv (z_1, ..., z_N)^T$. If the n-th unit is selected, then $z_n = 1$ and $z_m = 0$ for $m \neq n$. There are N configurations for the indicator variable $\mathbf{z}$ corresponding to the N units. The probability distribution for a complete event $(\mathbf{x}, \mathbf{z})$ in the MEF model is given by

$$P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \exp[\sum_{n=1}^{N} \sum_{\alpha=1}^{K} z_n r_\alpha(\mathbf{x}) \theta_{n,\alpha} + \sum_{n=1}^{N} z_n \theta_{n,0} + r_{K+1}(\mathbf{x}) - \Psi(\boldsymbol{\theta})], \tag{11}$$

where $\boldsymbol{\theta} \equiv \{\theta_{n,\alpha} | n = 1, ..., N; \alpha = 0, 1, ..., K\}$ is a set of MEF model parameters. The sufficient statistics of the MEF model are defined by $\mathbf{r}(\mathbf{x}, \mathbf{z}) \equiv \{z_n r_\alpha(\mathbf{x}) | n = 1, ..., N; \alpha = 0, 1, ..., K\}$, where $r_0(\mathbf{x}) \equiv 1$ is assumed. Using Eq. (3a) and (11), one can obtain the MEF model distribution for an observed event $\mathbf{x}$:

$$P(\mathbf{x}|\boldsymbol{\theta}) = \sum_{n=1}^{N} P(\mathbf{x}, n|\boldsymbol{\theta}) \tag{12a}$$

$$P(\mathbf{x}, n|\boldsymbol{\theta}) = \exp[\sum_{\alpha=1}^{K} r_\alpha(\mathbf{x}) \theta_{n,\alpha} + \theta_{n,0} + r_{K+1}(\mathbf{x}) - \Psi(\boldsymbol{\theta})] \ \ (n = 1, ..., N). \tag{12b}$$

$P(\mathbf{x}, n|\boldsymbol{\theta})$ represents a joint probability for a complete event such that the n-th unit is selected and the input $\mathbf{x}$ is observed.

For later convenience, let us define the Fisher information matrices $\mathbf{G}(\boldsymbol{\theta})$ and $\mathbf{V}(\boldsymbol{\phi})$ for the probability distributions $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})$ and $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))$, respectively:

$$G_{i,j}(\boldsymbol{\theta}) \equiv E\left[\left(\frac{\partial \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{\partial \theta_i}\right)\left(\frac{\partial \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{\partial \theta_j}\right)|\boldsymbol{\theta}\right] \tag{13a}$$

$$V_{i,j}(\boldsymbol{\phi}) \equiv E\left[\left(\frac{\partial \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial \phi_i}\right)\left(\frac{\partial \log P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial \phi_j}\right)|\boldsymbol{\theta}(\boldsymbol{\phi})\right] \tag{13b}$$

$$(i, j = 1, ..., M). \tag{13c}$$

The following relations are important for later calculations:

$$G_{i,j}(\boldsymbol{\theta}) = \frac{\partial^2 \Psi(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_j} = \frac{\partial \phi_i}{\partial \theta_j} \tag{14a}$$

$$V_{i,j}(\boldsymbol{\phi}) = \frac{\partial^2 H(\boldsymbol{\phi})}{\partial \phi_i \partial \phi_j} = \frac{\partial \theta_i}{\partial \phi_j} \tag{14b}$$

$$\mathbf{G}(\boldsymbol{\theta}) = \mathbf{V}(\boldsymbol{\phi})^{-1}. \tag{14c}$$

4 On-line EM algorithm

From a set of observed data $\mathbf{X} \equiv \{\mathbf{x}(t) \mid t = 1, ..., T\}$, the EFH model parameter $\boldsymbol{\theta}$ can be determined by the maximum likelihood method. In particular, the EM algorithm (Dempster, Laird and Rubin 1977) can be applied to the EFH model. The EM algorithm repeats the following E-step and M-step. It was proven that the likelihood for a set of observations increases (or does not change) after an E- and M-step. Therefore, the EM algorithm guarantees the convergence to a local maximum or a saddle point of the likelihood function.

- E (Estimation) step
  Let $\bar{\boldsymbol{\theta}}$ be the present estimator. By using $\bar{\boldsymbol{\theta}}$, the posterior probability of the hidden variable $\mathbf{z}$ for each observation $\mathbf{x}(t)$ is calculated according to the Bayes rule.

$$P(\mathbf{z}|\mathbf{x}(t),\bar{\boldsymbol{\theta}}) = P(\mathbf{x}(t),\mathbf{z}|\bar{\boldsymbol{\theta}})/P(\mathbf{x}(t)|\bar{\boldsymbol{\theta}}) \tag{15a}$$

$$P(\mathbf{x}(t)|\bar{\boldsymbol{\theta}}) = \sum_{\{\mathbf{z}\}} P(\mathbf{x}(t),\mathbf{z}|\bar{\boldsymbol{\theta}}). \tag{15b}$$

- M (Maximization) step
  By using the posterior probability (15), the expected log-likelihood $Q(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}},\mathbf{X})$ for the complete events is defined by

$$Q(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}},\mathbf{X}) = \sum_{t=1}^{T}\sum_{\{\mathbf{z}\}} P(\mathbf{z}|\mathbf{x}(t),\bar{\boldsymbol{\theta}}) \log P(\mathbf{x}(t),\mathbf{z}|\boldsymbol{\theta}). \tag{16}$$

  On the other hand, the log-likelihood of the observed data $\mathbf{X}$ is given by

$$L(\boldsymbol{\theta}|\mathbf{X}) = \sum_{t=1}^{T} \log P(\mathbf{x}(t)|\boldsymbol{\theta}) = \sum_{t=1}^{T} \log \left(\sum_{\{\mathbf{z}\}} P(\mathbf{x}(t),\mathbf{z}|\boldsymbol{\theta}) \right). \tag{17}$$

  Since an increase of $Q(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}},\mathbf{X})$ implies an increase of the log-likelihood $L(\boldsymbol{\theta}|\mathbf{X})$ (Dempster, Laird and Rubin 1977), $Q(\boldsymbol{\theta}|\bar{\boldsymbol{\theta}},\mathbf{X})$ is maximized with respect to the estimator $\boldsymbol{\theta}$. The stationary condition $\partial Q/\partial\boldsymbol{\theta} = 0$ can be solved as

$$\boldsymbol{\phi} = \langle \mathbf{r}(\mathbf{x},\mathbf{z}) \rangle (T). \tag{18}$$

  The symbol $\langle\cdot\rangle$ denotes a weighted mean with respect to the posterior probability (15) and is defined by

$$\langle \mathbf{r}(\mathbf{x},\mathbf{z}) \rangle (T) \equiv \frac{1}{T}\sum_{t=1}^{T}\sum_{\{\mathbf{z}\}} \mathbf{r}(\mathbf{x}(t),\mathbf{z}) P(\mathbf{z}|\mathbf{x}(t),\bar{\boldsymbol{\theta}}). \tag{19}$$

  The new model parameter $\boldsymbol{\theta}$ can be obtained by using Eq. (10):

$$\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\phi}) = \partial H(\boldsymbol{\phi})/\partial\boldsymbol{\phi}. \tag{20}$$

The above EM algorithm is based on batch learning, namely, the parameters are updated after seeing all of the observed data $\mathbf{X}$. In the following, we derive an on-line version of the EM algorithm. In the case of on-line learning, the observed data $\{\mathbf{x}(1),\mathbf{x}(2),...\}$ are supplied one at a time, and the estimator is changed after each observation. Let $\boldsymbol{\theta}(t)$ be the estimator after the t-th observation $\mathbf{x}(t)$ and introduce a time-dependent discount factor $\lambda(t)(0 \le \lambda(t) \le 1; t = 2,3,...)$. A discounted weighted mean is defined by

$$\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t) \equiv \eta(t) \sum_{\tau=1}^{t} \left(\prod_{s=\tau+1}^{t} \lambda(s) \right) \sum_{\{\mathbf{z}\}} \mathbf{r}(\mathbf{x}(\tau),\mathbf{z}) P(\mathbf{z}|\mathbf{x}(\tau),\boldsymbol{\theta}(\tau-1)) \tag{21a}$$

$$\eta(t) \equiv \left(\sum_{\tau=1}^{t} \prod_{s=\tau+1}^{t} \lambda(s) \right)^{-1} \qquad (t = 1,2,...). \tag{21b}$$

The discount factor $\lambda(t)$ is introduced for ignoring the effect of the old posterior values that employ the earlier inaccurate estimator. The schedule of the discount factor is discussed in Section 6. $\eta(t)$ is a normalization coefficient and plays the role of a learning rate. Replacing the weighted mean $\langle\cdot\rangle$ by the discounted weighted mean $\ll\cdot\gg$, the on-line EM algorithm is obtained.

The discounted weighted mean $\ll\cdot\gg$ can be calculated by a step-wise equation:

$$\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t) = \ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t-1) + \eta(t)\left[\sum_{\{\mathbf{z}\}} \mathbf{r}(\mathbf{x}(t),\mathbf{z})P(\mathbf{z}|t) - \ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t-1)\right] \quad (22a)$$

$$\eta(t) = (1+\lambda(t)/\eta(t-1))^{-1}, \quad (22b)$$

where $P(\mathbf{z}|t) \equiv P(\mathbf{z}|\mathbf{x}(t),\boldsymbol{\theta}(t-1))$. Subsequently, the new parameters are calculated as

$$\boldsymbol{\phi}(t) = \ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t) \quad (23a)$$

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}(\boldsymbol{\phi}(t)) = \frac{\partial H(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}|_{\boldsymbol{\phi}(t)}. \quad (23b)$$

Equations (22) and (23) define the on-line EM algorithm for the EFH model.

5 Stochastic approximation

If an infinite number of data, which are drawn independently according to the input data distribution $\rho(\mathbf{x})$, is available, the log-likelihood function is given by

$$L(\boldsymbol{\theta}) = E[\log \sum_{\{\mathbf{z}\}} P(\mathbf{x},\mathbf{z}|\boldsymbol{\theta})]_\rho, \quad (24)$$

where $E[\cdot]_\rho$ denotes the expectation value with respect to the input data distribution $\rho(\mathbf{x})$. The maximum likelihood condition $\partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta} = 0$ turns out to be

$$\boldsymbol{\phi} = E[\sum_{\{\mathbf{z}\}} \mathbf{r}(\mathbf{x},\mathbf{z})P(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})]_\rho \quad (25a)$$

$$\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\phi}) = \partial H(\boldsymbol{\phi})/\partial\boldsymbol{\phi}. \quad (25b)$$

The on-line EM algorithm, Eq. (22) and (23), can be written as

$$\Delta\boldsymbol{\phi}(t) \equiv \boldsymbol{\phi}(t) - \boldsymbol{\phi}(t-1)$$
$$= \eta(t)[\sum_{\{\mathbf{z}\}} \mathbf{r}(\mathbf{x}(t),\mathbf{z})P(\mathbf{z}|\mathbf{x}(t),\boldsymbol{\theta}(t-1)) - \boldsymbol{\phi}(t-1)] \quad (26a)$$

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}(\boldsymbol{\phi}(t)) = \frac{\partial H(\boldsymbol{\phi})}{\partial\boldsymbol{\phi}}|_{\boldsymbol{\phi}(t)}. \quad (26b)$$

Using the log-likelihood for the current data $\mathbf{x}(t)$,

$$L(\mathbf{x}(t)|\boldsymbol{\theta}) \equiv \log(\sum_{\{\mathbf{z}\}} P(\mathbf{x}(t),\mathbf{z}|\boldsymbol{\theta})), \quad (27)$$

the update rule (26) is further rewritten as

$$\Delta\boldsymbol{\phi}(t) = \eta(t)\left(\frac{\partial L(\mathbf{x}(t)|\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)|_{\boldsymbol{\theta}(t-1)} \quad (28a)$$

$$= \eta(t)\left(\frac{\partial\boldsymbol{\phi}}{\partial\boldsymbol{\theta}}\right)\left(\frac{\partial L(\mathbf{x}(t)|\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial\boldsymbol{\phi}}\right)|_{\boldsymbol{\phi}(t-1)} \quad (28b)$$

$$= \eta(t)(\mathbf{V}(\boldsymbol{\phi}(t-1)))^{-1}\left(\frac{\partial L(\mathbf{x}(t)|\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial\boldsymbol{\phi}}\right)|_{\boldsymbol{\phi}(t-1)}, \quad (28c)$$

where the relation (14) is used.

Equation (28) shows that the on-line EM algorithm (26) is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix $\mathbf{V}^{-1}$ as a coefficient matrix. In order to apply the stochastic approximation theory (Kushner and Yin 1997), it is assumed that the expectation parameter $\boldsymbol{\phi}(t)$ is bounded in a compact region $|\phi_i(t)| \leq \phi_{max}(i = 1, ..., M)$ for a sufficiently large constant ϕ_{max}. This can be achieved by a simple procedure where $\phi_i(t)$ is set to $\phi_{max}(-\phi_{max})$, if $\phi_i(t)$ becomes larger (smaller) than $\phi_{max}(-\phi_{max})$. In numerical calculations, this kind of procedure is often necessary to avoid numerical explosion. It is also assumed that the input data distribution $\rho(\mathbf{x})$ has a compact support or that $\rho(\mathbf{x})$ decreases exponentially as $|\mathbf{x}|$ goes to infinity. This is also a natural assumption because actual data are always distributed in a finite domain. These assumptions guarantee that

$$E[|\mathbf{V}^{-1}(\boldsymbol{\phi})(\partial L(\mathbf{x}|\boldsymbol{\theta}(\boldsymbol{\phi}))/\partial\boldsymbol{\phi})|^2]_\rho < \infty. \tag{29}$$

The effective learning rate $\eta(t)(\geq 0)$ is assumed to satisfy the condition

$$\sum_{t=1}^{\infty} \eta(t) = \infty \quad and \quad \sum_{t=1}^{\infty} \eta^2(t) < \infty. \tag{30}$$

Under the above conditions, the stochastic approximation theory (Kushner and Yin 1997) guarantees that the asymptotic behavior of the estimator satisfying (28) is described by a continuous time differential equation

$$\frac{d\boldsymbol{\phi}}{dt}(t) = (\mathbf{V}(\boldsymbol{\phi}(t)))^{-1} \left(\frac{\partial L(\boldsymbol{\theta}(\boldsymbol{\phi}))}{\partial \boldsymbol{\phi}} \right) |_{\boldsymbol{\phi}(t)}, \tag{31}$$

and that $\boldsymbol{\phi}(t)$ converges to a limit point of the differential equation (31) or converges to a boundary point of the compact region. Equation (31) has the Lyapunov function $(-L(\boldsymbol{\theta}(\boldsymbol{\phi})))$, which always decreases its value over time, because the Fisher information matrix and its inverse are positive definite matrices. Therefore, the estimator converges to a local maximum of the likelihood function $L(\boldsymbol{\theta}(\boldsymbol{\phi}))$ or converges to a boundary point of the compact region. In addition to the above assumptions, if $(\mathbf{V}^{-1}(\boldsymbol{\phi})(\partial L(\boldsymbol{\theta}(\boldsymbol{\phi}))/\partial\boldsymbol{\phi}))$ at the boundary of the compact region always points inward, the estimator $\boldsymbol{\phi}(t)$ converges to a local maximum of the likelihood function inside the compact region with probability one.

Equation (28c) has a similar form as the natural gradient method proposed by Amari (1998), which gives the optimal asymptotic convergence. However, there is an important difference. The Fisher information matrix in (28c) is defined for the probability distribution of a complete event $P(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}(\boldsymbol{\phi}))$. On the other hand, the Fisher information matrix in the natural gradient method is defined for the probability distribution of an observed event $P(\mathbf{x}|\boldsymbol{\theta}(\boldsymbol{\phi}))$. It is beyond the scope of the current paper to study the relationship between the on-line EM algorithm and the natural gradient method.

6 Schedule of Discount Factor

The performance of the on-line EM algorithm depends on the schedule of the effective learning rate $\eta(t)$, which is controlled by the schedule of the discount factor $\lambda(t)$. In this section, the schedule of the discount factor $\lambda(t)$ is discussed in detail. It is convenient to introduce new parameters by

$$\kappa(t) \equiv \sum_{\tau=1}^{t} \prod_{s=\tau+1}^{t} \lambda(s) = 1/\eta(t) \tag{32a}$$

$$\epsilon(t) \equiv 1 - \lambda(t). \tag{32b}$$

$\kappa(t)$ represents an effective number of averaged data that contributes to the discounted weighted mean $\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t)$ defined by Eq. (21). In particular, $\kappa(t) = t$ for $\lambda(t) = 1$. If $\lambda(t) = \lambda(const.)$, then $\lambda^t = (1-\epsilon)^t \simeq 0$ for $t > (1/\epsilon)$. Therefore, $(1/\epsilon)$ corresponds to an effective length of the memory window. The contribution from the past data earlier than this window length can be neglected in the calculation of the discounted weighted mean $\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t)$.

From the definition (32), $\kappa(t)$ takes the maximum and minimum values for $\lambda(t) = 1$ and $\lambda(t) = 0$, respectively. Therefore, the following relations hold.

$$1 \leq \kappa(t) \leq t \tag{33a}$$

$$1 \geq \eta(t) \geq 1/t. \tag{33b}$$

If this constraint is satisfied, Eq. (22b) can be solved for $\lambda(t)$ as

$$\lambda(t) = \eta(t-1)(1/\eta(t) - 1). \tag{34}$$

Therefore, one can determine the effective learning rate $\eta(t)$ without referring to the discount factor $\lambda(t)$ as long as the constraint (33) is satisfied. In practice, however, it is easier to control the discount factor $\lambda(t)$ than the effective learning rate $\eta(t)$ as will be explained later.

In order to see asymptotic behavior of the effective learning rate $\eta(t)$, the recursive equation (22b) is rewritten as

$$\kappa(t) - \kappa(t-1) = 1 - \epsilon(t)\kappa(t-1). \tag{35}$$

If $\epsilon(t) \to t^{-\beta}(\beta > 1)$ as $t \to \infty$, then $(\kappa(t) - \kappa(t-1)) \to 1$, is satisfied. This implies that $\kappa(t) \to t + o(1)$, i.e.,

$$\eta(t) \stackrel{t\to\infty}{\longrightarrow} 1/t \quad for \quad \epsilon(t) \stackrel{t\to\infty}{\longrightarrow} t^{-\beta}(\beta > 1). \tag{36}$$

If $\epsilon(t) \to 1/(\gamma t)$ and $\kappa(t) \to (\Gamma t)$, then the relation, $\Gamma = 1 - \Gamma/\gamma$, can be derived from (35) in the limit $t \to \infty$, i.e.,

$$\eta(t) \stackrel{t\to\infty}{\longrightarrow} 1/(\Gamma t) \quad for \quad \epsilon(t) \stackrel{t\to\infty}{\longrightarrow} 1/(\gamma t), \tag{37}$$

where Γ and γ are related by

$$0 < \Gamma = \gamma/(\gamma+1) < 1. \tag{38}$$

If $\eta(t) \stackrel{t\to\infty}{\longrightarrow} 1/(\Gamma t)$ is satisfied, the stochastic approximation condition (30) is satisfied.

In the experiments, we used a (1/t)- schedule for $\epsilon(t)$:

$$\epsilon(t) = \frac{1}{(t-2)\gamma + 1/\epsilon_0}, \tag{39}$$

where $\epsilon(2) = \epsilon_0$ is the initial value of $\epsilon(t)$ and γ determines the decay rate of $\epsilon(t)$. If the initial value of $\eta(t)$ is given by[1]

$$\eta(1) = \eta_0 = \epsilon_0(1+\gamma), \tag{40}$$

the effective learning rate $\eta(t)$ also satisfies a (1/t)- schedule,

$$\eta(t) = \frac{1}{(t-1)\Gamma + 1/\eta_0}, \quad \Gamma = \gamma/(\gamma+1). \tag{41}$$

[1] In the definition of $\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t)$ and $\eta(t)$, (21), it is assumed that there is no initial value contribution for $\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t)$. In the following arguments, we assume that there is an initial value contribution for $\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (t)$, i.e., $\ll \mathbf{r}(\mathbf{x},\mathbf{z}) \gg (0) \equiv \phi(0)$ and $\eta(1) = \eta_0$. This is equivalent to assuming that we start at $t = t_0 > 1$ and $\phi(t_0 - 1)$ is regarded as the initial value.

On the other hand, if the initial value does not satisfy Eq. (40), $\eta(t)$ does not satisfy the (1/t)- schedule (41). In the experiments, we used three independent constants, ϵ_0, η_0 and γ, to control the learning schedule. They have clear physical meaning. The initial condition $\eta(1) = \eta_0$ implies that the initial value for the expectation parameter $\boldsymbol{\phi}(0)$ can be regarded as an average of $(1/\eta_0 - 1)$ data points. Therefore, the value of η_0 is determined according to the uncertainty of the initial estimator $\boldsymbol{\phi}(0)$. The initial condition $\epsilon(2) = \epsilon_0$ implies that the effective length of the memory window in the early stage of learning is given by $(1/\epsilon_0)$. The decay rate γ controls the asymptotic behavior of $\eta(t)$ as in Eq. (37) and (38). Fast and stable learning can be obtained by a schedule with $\eta_0 \sim o(1)$, $0 < \epsilon_0 \ll 1$ and $0 < \gamma < 1$. There are three phases which are characterized by different $\eta(t)$ behaviors (Figure 2). In the first phase, $1 \leq t \leq 1/\epsilon_0$, $\epsilon(t)$ is nearly equal to ϵ_0 and $\eta(t)$ rapidly decreases to ϵ_0. In this period, a rough estimate of the parameters takes place very quickly. This period can be considered an initialization phase. If the initial value for $\boldsymbol{\phi}(0)$ is a good estimator, this period is not necessary and it is possible to use a (1/t)- schedule for $\eta(t)$, (41),in which η_0 is given by $\eta_0 = \epsilon_0(1+\gamma) \ll 1$. In the second phase, $1/\epsilon_0 \leq t \leq 1/(\epsilon_0\gamma)$, $\epsilon(t)$ and $\eta(t)$ are nearly equal to ϵ_0. This period is the main part of the learning process, in which a fairly good estimator is obtained. In the third phase, $1/(\epsilon_0\gamma) \leq t$, $\eta(t)$ behaves like $1/(\Gamma t)$ in Eq. (37). This period stabilizes the estimator value by the (1/t)- schedule of $\eta(t)$.

In stochastic approximation theory, a schedule $\eta(t) = 1/t$ is often used. This corresponds to $\lambda(t) = 1$ in our on-line EM algorithm, i.e., there is no discount effect. Therefore, the effects of the old posterior values employing the earlier inaccurate estimator do not diminish and the initial parameter value greatly affects the learning performance. Nevertheless, asymptotic convergence to a local maximum of the likelihood function is guaranteed because contributions from an infinite number of data in the later stage of learning overwhelm the earlier finite number of contributions employing an inaccurate estimator.

7 Experiment

7.1 Mixture of Gaussian Model

The performance of the on-line EM algorithm was examined by using the Mixture of Gaussian (MG) model. An MG model for the 2-dimensional input vector $\mathbf{x} \equiv (x_1, x_2)$ is defined by

$$P(\mathbf{x}|\boldsymbol{\theta}) = \sum_{n=1}^{N} P(n) \exp[-|\mathbf{x} - \mathbf{m}_n|^2/(2\sigma_n^2)](2\pi\sigma_n^2)^{-1}. \tag{42}$$

The center and the variance of the n-th Gaussian function are denoted by $\mathbf{m}_n$ and σ_n^2, respectively. A set of model parameters is given by $\boldsymbol{\theta} \equiv \{\theta_{n,0}, \mathbf{m}_n, \sigma_n^2 | n = 1, ..., N\}$, where $P(n) \equiv e^{\theta_{n,0}}/(\sum_{m=1}^{N} e^{\theta_{m,0}})$.

For a data generation model, an MG model consisting of four Gaussian functions was prepared. The centers and the variances of the model are shown in Fig. 1A. The mixing constant was given by $(P(1), ..., P(4)) = (4/9, 2/9, 2/9, 1/9)$. Three training data sets consisting of 100, 1,000 and 10,000 points were randomly generated according to the data generation model distribution $\rho(\mathbf{x})$. 100 and 1,000 data points in the training data sets are shown in Figs. 1A and 1B, respectively. In order to measure the generalization error, the Kullback-Leiber (KL) divergence from the data generation distribution $\rho(\mathbf{x})$ to the MG model distribution $P(\mathbf{x}|\boldsymbol{\theta})$ was calculated:

$$K(\rho||P) \equiv \int dx_1 dx_2 \rho(\mathbf{x}) \log(\rho(\mathbf{x})/P(\mathbf{x}|\boldsymbol{\theta})). \tag{43}$$

This was evaluated on a 51 x 51 point mesh grid.

The MG model was trained by using three algorithms: the on-line EM, the batch EM and the on-line gradient ascent algorithms. The on-line gradient ascent algorithm is defined by

$$\Delta\boldsymbol{\theta}(t) = \eta(t) \left(\frac{\partial L(\mathbf{x}(t)|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right) |_{\boldsymbol{\theta}(t-1)}. \tag{44}$$

For each learning method and each dataset, learning processes starting from 20 different initial conditions were performed. The initial conditions were generated as follows. First, the initial center positions were randomly generated from the region [0 1] x [0 1]. The initial variances for all the units were set to the variance among the initial centers. The mixing constants were set to the same value, (1/N). The same set of initial conditions was used for all experiments.

In one epoch, each training algorithm sees all training data once. The batch EM algorithm updates the model parameters once at the end of each epoch, while the on-line algorithms update them for each datum. The computational costs in one epoch for three algorithms are on the same order that is proportional to (N x (Size of training data)). The batch EM algorithm can be considered a special type of the on-line EM algorithm, in which the model parameters are only updated at the end of each epoch, and the discount factor is given by $\lambda(t) = 0$ at the beginning of each epoch, or $\lambda(t) = 1$ otherwise. This fact also supports the assumption that the learning speed of each method can be compared in terms of number of epochs.

7.2 Schedule of Discount Factor

In order to achieve fast and stable learning performance, we used a (1/t)- schedule for $\epsilon(t)$, (39). In the current experiments, the initial parameter value was randomly chosen. Therefore, we set $\eta_0 = 0.5$, which means that the initial parameter is regarded as a contribution from one data point. The initial value ϵ_0 is set to 0.01, which means that a weighted average of the last 100 data points are taken in the early stage of learning. The decay rate γ is set to 0.05. The performance of the on-line EM algorithm in the current experiments are fairly robust for the variation of these constants if $\eta_0 \sim o(1)$, $0.001 \leq \epsilon_0 \leq 0.01$ and $0.01 \leq \gamma \leq 0.1$ are satisfied.

The time course of $\eta(t)$ for various schedules are plotted in Fig. 2. A schedule $\eta(t) = 1/t$, which corresponds to $\eta_0 = 1$ and $\epsilon(t) = 0$, exhibits a fast decrease of $\eta(t)$. The learning curves for this schedule are similar to those for the batch EM algorithm shown in Fig. 4. In a (1/t)- schedule for $\eta(t)$, (41), with $\eta_0 = 0.5$ and $\gamma = 0.1$ ($\epsilon_0 = \eta_0/(1+\gamma) \simeq 0.45$), $\eta(t)$ decreases very slowly in the early stage of learning. The learning performance in this schedule is not good because the estimator forgets the contribution of past data too fast. The (1/t)- schedules for $\epsilon(t)$, (39), with $\eta_0 = 0.5$, $\epsilon_0 = 0.01$, and $\gamma = \{0.1, 0.05, 0.01\}$ exhibit a fast decrease of $\eta(t)$ in the early stage of learning and a slow decrease of $\eta(t)$ in the later stage. These schedules worked well in the experiments.

If the initial parameter value is a good estimator, one can use a (1/t)- schedule for $\eta(t)$ with a small initial value of $\eta_0 = \epsilon_0(1+\gamma) \simeq \epsilon_0 \ll 1$.

7.3 Results

The learning curves for the on-line EM algorithm are shown in Fig. 3. The estimator quickly converged close to the optimal value in all cases. It nearly converged before 20,000 data points were presented, regardless of the training data size. The final generalization error improved as the amount of training data increased. Although over-training can be seen for the data set with 100 data (Fig. 3A), the generalization error at the asymptotic regime is the same as that obtained by the batch EM algorithm (see later discussion). The MG model with 20 units has an excessive degree of freedom. Therefore, there are a continuum set of optimal estimators and a continuum set of suboptimal estimators. This explains the small variations in the generalization error for the case N=20 (Fig. 3D).

The learning curves for the batch EM algorithm are shown in Fig. 4. Surprisingly, the learning speed of the batch EM algorithm was much slower than that of the on-line EM algorithm. The final generalization errors improved as the amount of training data increased and they were almost the same as those obtained by the on-line EM algorithm, except in the exceptional cases described below. The learning speed, however, did not improve even if the amount of training data increased. The estimator of an MG model with N=4 was trapped to a local maximum of the likelihood function for some initial conditions (Figs. 4A-4C). The increase in the training data size did not help this situation. The configuration of this local maximum solution is shown in Fig. 1C. In this solution, two units try to approximate one Gaussian function, so that the remaining two units must try to approximate three Gaussian functions. The estimator trained by the on-line EM algorithm nearly converged to the global maximum likelihood estimator, even when the same initial condition was used. The solution obtained by the on-line EM algorithm is shown in Fig. 1D. If an MG model with 20 units was used, the estimator seemed to converge to suboptimal estimators (Fig. 4D), as in the case of the on-line EM algorithm.

The reason for the slow learning and the local maximum problem can be interpreted as follows. In the early stage of the batch EM learning, the same posterior probability employing an inaccurate estimator is used for all training data in one epoch. This may cause an inappropriate assignment of the units for the training data and result in a poor estimation of the new parameter. In the on-line EM algorithm, the posterior probability is calculated with the new parameter, which is updated each time data is presented. Therefore, the unit assignment gradually improves as more data are presented.

The learning curves for the on-line gradient ascent algorithm are shown in Fig. 5, in which the best learning rate schedule is used. The learning performance of the on-line gradient ascent algorithm was very sensitive to the learning rate schedule. If the initial value for the learning rate η_0 was not smaller than 0.001, unstable oscillatory behavior occurred in the early stage of learning. The best performance was obtained by a (1/t)- schedule for $\eta(t)$, (41), with $\eta_0 = 0.0001$ and $\gamma = 0.1$, (i.e., $\epsilon_0 = \eta_0/(1+\gamma) \simeq 0.00009$, $\Gamma = \gamma/(1+\gamma) \simeq 0.09$). The performance for the fixed learning rate, $\eta(t) = \eta_0 = 0.0001$, was slightly worse than the above best performance. In order to achieve the same accuracy of the generalization error obtained by the on-line EM algorithm, 100 times more epochs are needed in the on-line gradient ascent algorithm for all cases in Fig. 5.

8 Conclusion

In this article, an on-line EM algorithm was derived for general Exponential Family models with Hidden variables (EFH models). It was proven that the on-line EM algorithm is equivalent to a stochastic gradient method with the inverse of the Fisher information matrix as a coefficient matrix. As a result, the stochastic approximation theory guarantees the convergence to a local maximum of the likelihood function.

The performance of the on-line EM algorithm was examined by using the mixture of Gaussian model. The simulation results showed that the on-line EM algorithm was much faster than the batch EM algorithm and the on-line gradient ascent algorithm. The efficiency of the on-line EM algorithm over the batch EM algorithm became more prominent as the amount of training data increased. In addition, the on-line EM algorithm could escape from a local maximum, even when the batch EM algorithm was trapped to a local maximum solution. The inherent stochastic nature of the on-line EM algorithm in the early stage of learning may be the reason for this fact. This also shows the superiority of the on-line EM algorithm over the batch EM algorithm, although this does not guarantee the convergence to the global maximum.

One of the advantages of our on-line EM algorithm is that the estimator at each time step can be represented by the discounted weighted mean $\ll \mathbf{r}(\mathbf{x}, \mathbf{z}) \gg (t)$ defined by Eq. (21). Therefore,

the on-line EM algorithm gives a reasonable estimator even in the early stage of learning. Moreover, the learning rate schedule can be systematically designed and the fast learning schedule, in which $\eta_0 \sim o(1)$, is possible. This kind of fast learning schedule is not possible for the on-line gradient ascent algorithm because it causes unstable oscillatory learning behaviors.

We have pointed out that the on-line EM algorithm has a similar form as the natural gradient method proposed by Amari (1998), which gives the optimal asymptotic convergence. The inverse of the Fisher information matrix in the on-line EM algorithm may contribute to fast learning performance. In our on-line EM algorithm, however, it is not necessary to calculate the inverse of the Fisher information matrix. In the future, it would be interesting to study the relation of our algorithm to the natural gradient method.

References

[1] Amari, S. (1967). Theory of adaptive pattern classifiers. *IEEE Trans. EC*, **16**, 299-307.

[2] Amari, S. (1985),*Differential geometrical method in statistics*, Springer Lecture Notes in Statistics,**28**, Springer.

[3] Amari, S. (1995). Information Geometry of the EM and em Algorithms for Neural Networks. *Neural Networks*, **9**, 1379-1408.

[4] Amari, S. (1998), Natural Gradient Works Efficiently in Learning, *Neural Computation*, **10**, pp.251-276.

[5] Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press. New York.

[6] Bottou, L., (1999). On-line learning and stochastic approximations. in *On-line Learning and Neural Networks*, Saad, D. (ed.), Cambridge University Press, UK.

[7] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, **39**, 1-22.

[8] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, **3**, 79-87.

[9] Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, **6**, 181-214.

[10] Kushner, H. J., & Yin, G. G. (1997). *Stochastic Approximation Algorithms and Applications*, New York: Springer-Verlag.

[11] Murata, N. (1999). A statistical study on on-line learning. in *On-line Learning and Neural Networks*, Saad, D. (ed.), Cambridge University Press, UK.

[12] Neal, R. M., & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. To appear in M. I. Jordan (Ed.), *Learning in Graphical Models*, Kluwer Academic Press.

[13] Nowlan, S. J. (1991). Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures. *CMU Technical Report*, **CS-91-126**, Pittsburgh: Carnegie Mellon University.

[14] Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, **22**, 400-407.

[15] Sato, M., & Ishii, S. (1999). On-line EM Algorithm for the Normalized Gaussian Network. To appear in *Neural Computation*.

[16] Xu, L., Jordan, M. I., & Hinton, G. E. (1995). An alternative model for mixtures of experts. In G. Tesauro, D. S. Touretzky, & T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7* (pp. 633-640), Cambridge, MA: MIT Press.

- **Figure 1**
  (A), (B): Centers and variances of the data generation model consisting of four Gaussian functions are shown by four circles. The center and radius of a circle correspond to the center and standard deviation of a Gaussian function, respectively. 100 and 1,000 training data points are plotted in (A) and (B), respectively. (C): Local maximum solution obtained by the batch EM algorithm is shown by four circles, which correspond to four Gaussian functions in the trained MG model. (D): Solution near the global maximum likelihood estimator obtained by the on-line EM algorithm is shown by four circles. The same initial condition is used in (C) and (D).

- **Figure 2**
  The time courses of the effective learning rate $\eta(t)$ for various schedules are plotted. Dashed and dash-dotted lines represent a schedule $\eta(t) = 1/t$ and a (1/t)- schedule for $\eta(t)$, (41), with $\eta_0 = 0.5$, and $\gamma = 0.1$, respectively. Solid lines represent the (1/t)- schedules for $\epsilon(t)$, (39), with $\eta_0 = 0.5$, $\epsilon_0 = 0.01$, and $\gamma = \{0.1, 0.05, 0.01\}$.

- **Figure 3**
  Learning curves for the on-line EM algorithm. Ordinate denotes the generalization error measured by the KL-divergence from the data generation model distribution to the trained MG model probability distribution. Abscissa denotes the epoch number. Each figure plots learning curves starting from 20 different initial conditions. The sizes of the training data are 100 (A); 1,000 (B); 10,000 (C); and 1,000 (D). The numbers of units in the trained MG models are four (A, B, and C) and 20 (D).

- **Figure 4**
  Learning curves for the batch EM algorithm. The training data sets and the trained MG models are the same as in Figure 3.

- **Figure 5**
  Learning curves for the on-line gradient ascent algorithm. The training data sets and the trained MG models are the same as in Figure 3.

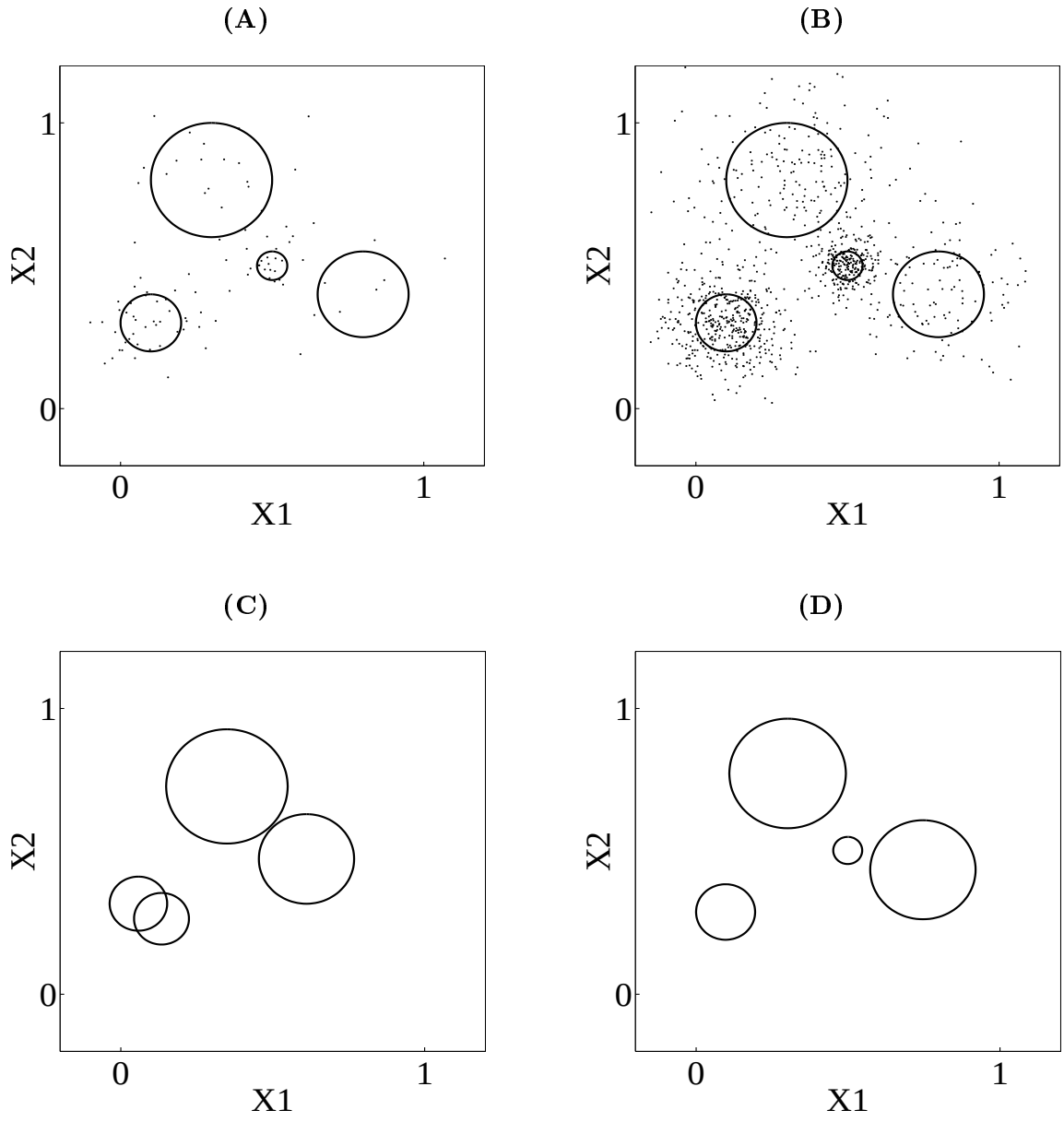

Figure 1

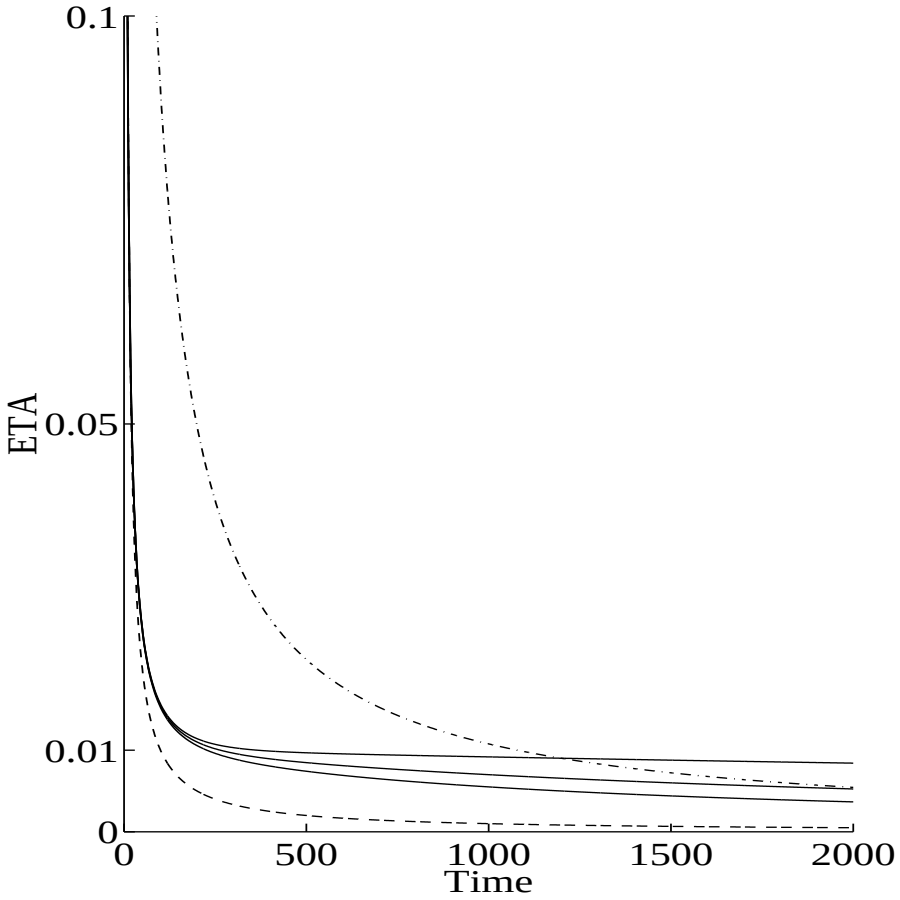

Figure 2

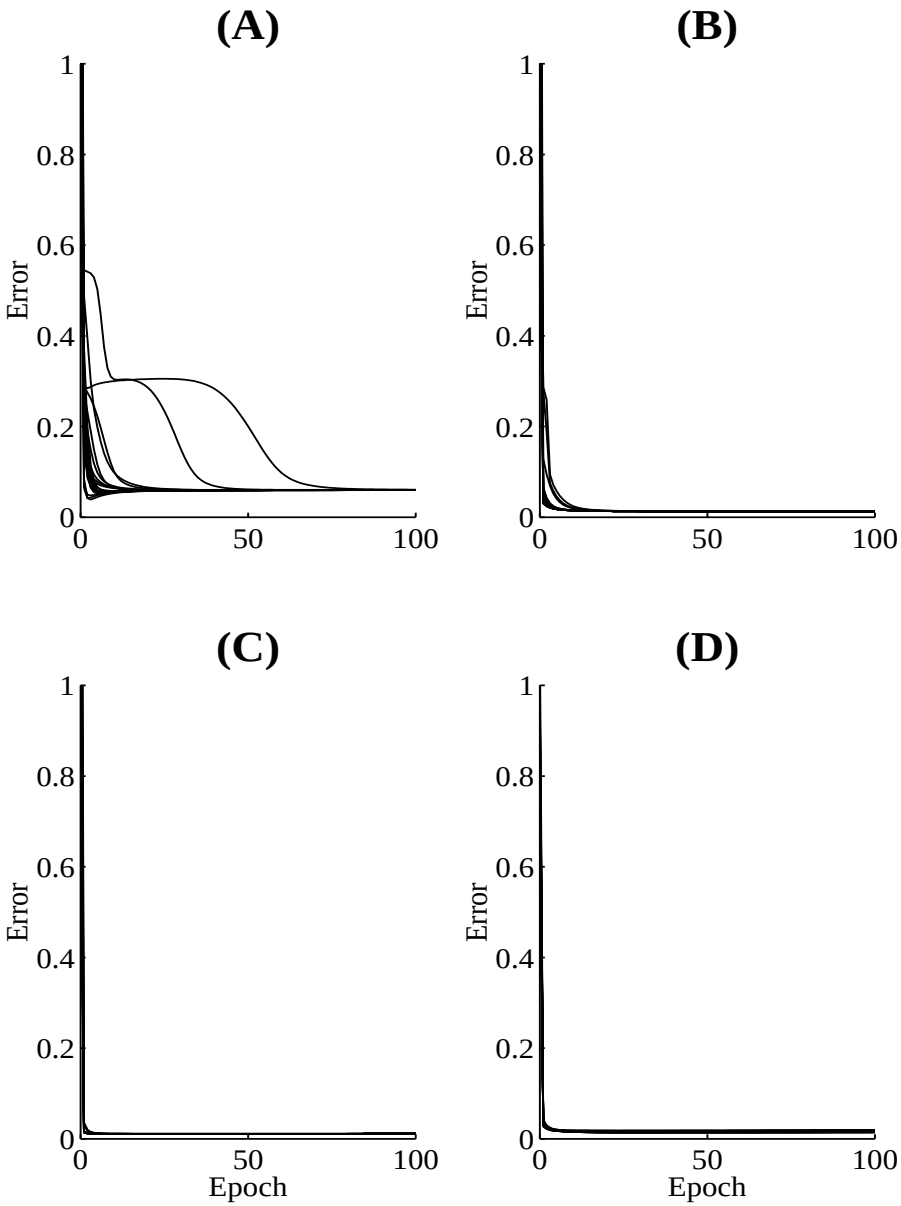

Figure 3

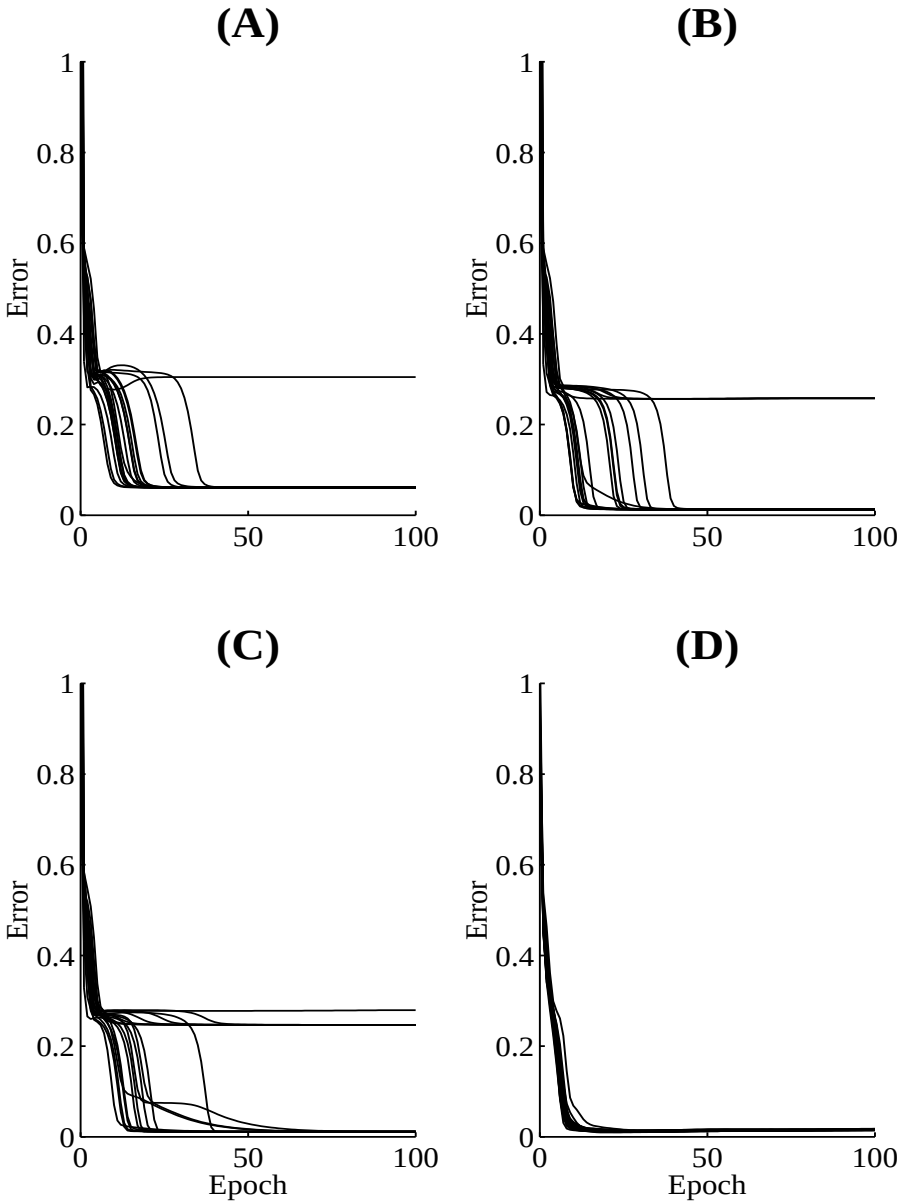

Figure 4

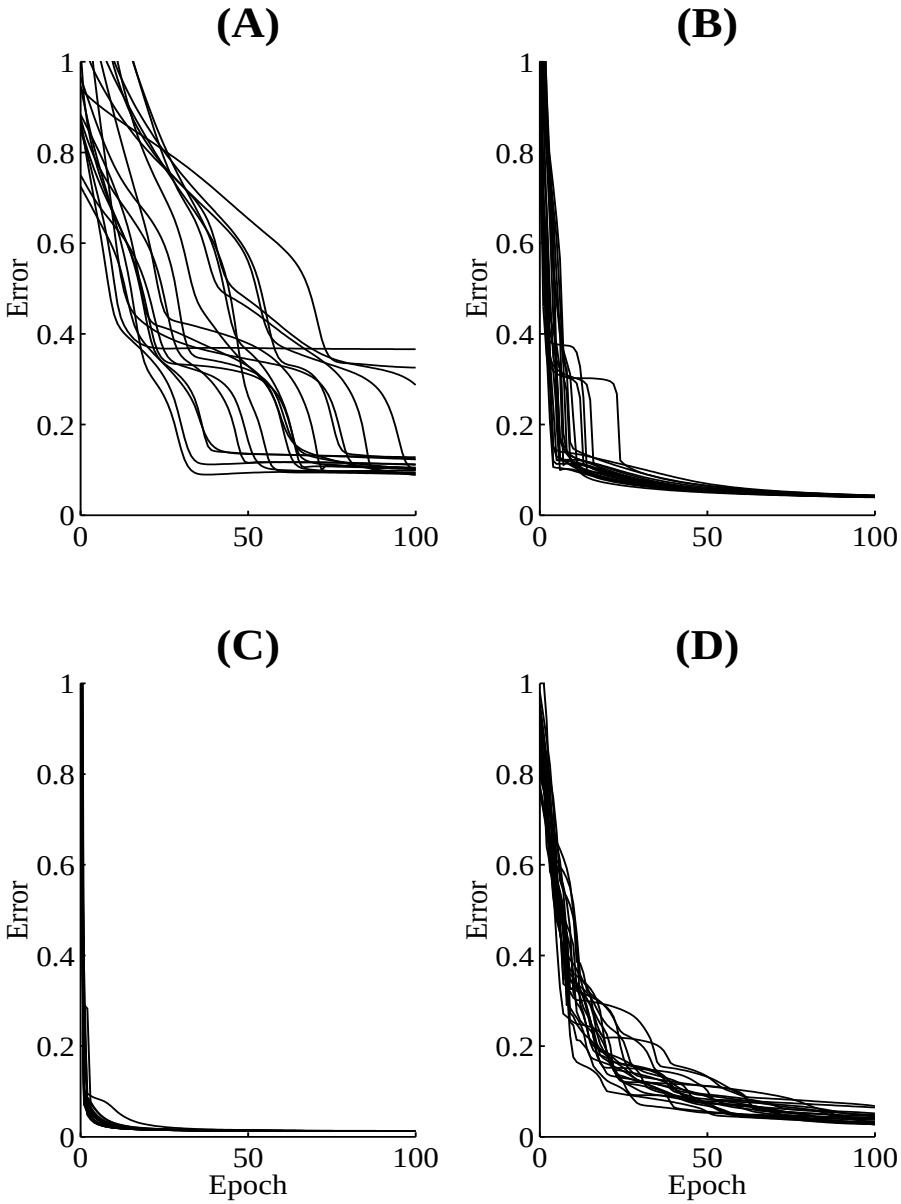

Figure 5